УДК 519.83

О.А. Емец, Д.Н. Ольховский, Е.В. Ольховская
Полтавский университет экономики и торговли, Украина
Украина, 36014, г. Полтава, ул. Коваля, 3

Сравнение методов решения игровых задач: числовые эксперименты

O.O. Iemets, D.M. Olhovskiy, O.V. Olhovskaya
Poltava University of economics and trade, Ukraine
Ukraine, 36014, c. Poltava, Kovalya st., 3

Comparison of Methods for Solving Game Tasks: Numerical Experiments

О.О. Ємець, Д.М. Ольховський, О.В. Ольховська
Полтавський університет економіки і торгівлі, Україна
Україна, 36014, м. Полтава, вул. Коваля, 3

Порівняння методів розв’язування ігрових задач: числові експерименти

В статье приведены результаты числовых экспериментов по практической эффективности метода Брауна-Робинсон и монотонного алгоритма для матричных игр с точки зрения скорости сходимости и временных затрат, так и их точности.
Ключевые слова: числовые эксперименты, метод Брауна-Робинсон, матричные игры.

The paper presents the results of numerical experiments about practical effectiveness of the Brown- Robinson method and monotonic algorithm for matrix games in terms rate of convergence and exactness time outlay.
Key words: numerical experiments, Brown-Robinson method, matrix games.

У статті представлені результати числових експериментів щодо практичної ефективності метода Брауна-Робінсон і монотонного алгоритму для матричних ігор з точки зору збіжності і часових обмежень та точності.
Ключові слова: числові експерименти, метод Брауна-Робінсон, матричні ігри.

Анализ конфликтных ситуаций приводит к возникновению новых моделей игровых задач. В работах [1-9] введен новый класс задач комбинаторной оптимизации игрового типа с двумя игроками, в которых платежи представлены матрицей, а на стратегии игроков накладываются комбинаторные ограничения. Такие задачи имеют широкое применение в разных областях человеческой деятельности. В [5] предложен приближенный итерационный метод решения таких задач, в основе которого лежат те же идеи, что и в основании метода Брауна-Робинсон [10]. Как известно, метод Брауна-Робинсон показал определенную практическую эффективность, но, на отдельных классах задач, как отмечено в [11] не достаточную скорость сходимости. Учитывая это для решения задач, рассмотренных в [1-9], возникает необходимость разработки других методов для решения матричных игр, которые будут основаны на других подходах.

В [11] предложен монотонный метод решения игровых задач, который, по мнению авторов этой публикации, является более эффективным и имеет лучшую

скорость сходимости в сравнении с методом Брауна-Робинсон. В данной статье поставлена задача на основе числовых экспериментов сравнить известные методы решения матричных игр как с точки зрения скорости сходимости и временных затрат, так и их точности, чтобы использовать эту модификацию при решении задачи из [1-9].

Постановка задачи

Рассмотрим классическую матричную игру [12], в которой первый игрок имеет m стратегий $i=\overline{1,m}$, второй игрок – n стратегий $j=\overline{1,n}$. Игроки поочередно делают ходы: первый игрок выбирает свою i-ю стратегию, второй – свою j-ю стратегию, после чего первый игрок получает выигрыш a_{ij} за счет второго игрока (если $a_{ij}<0$, то это означает, что первый игрок платит второму сумму $|a_{ij}|$).

Значение функции выигрыша можно записать в виде прямоугольной матрицы A:

$$A=\begin{pmatrix} a_{11} & \ldots & \ldots & a_{1n_1} \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & \ldots & \ldots & a_{mn} \end{pmatrix}.$$

Таким образом, рассматривается матрица $A=(a_{ij})$, $i=\overline{1,m}$, $j=\overline{1,n}$ размерности $(m\times n)$. Проведение каждой партии матричной игры с матрицей A сводится к выбору первым игроком i-й строки, а вторым игроком j-го столбца, и получения первым игроком (за счет второго) выигрыша a_{ij}.

Алгоритмы методов

Если размерность исходной матрицы небольшая, то такую игру, как известно, можно свести к поиску седловой точки функции или к получению решения пары двойственных задач линейного программирования [12]. Матричные игры, в которых исходная матрица имеет довольно большие размерности, чаще всего решаются приближенными итерационными методами, в частности, методом Брауна-Робинсон [10], и монотонным итеративным алгоритмом [11] и др. Рассмотрим алгоритмы названных методов.

Метод Брауна-Робинсон

Идея метода – многоразовое фиктивное разыгрывание игры с заданной матрицей $A=(a_{ij})$. Один розыгрыш игры называют итерацией, количество которых неограниченно. С ростом числа итераций N смешанные частоты применения чистых стратегией игроков приближаются к их смешанным оптимальным стратегиям.

Расчеты делаются в предположении, что игроки хотят увеличить свой выигрыш (уменьшить проигрыш). Предполагается, что они не знают своих оптимальных стратегий. Ходы игроки делают в соответствии с принципом: будущее похоже на прошлое, учитывая все сделанные итерации.

Первая стратегия (например, первым игроком) выбирается произвольно – возможно с целью увеличить возможный выигрыш.

Выбор следующего (s-го) хода 2-го игрока (после s ходов 1-го игрока) происходит выбором стратегии j_s, такой, что $\sum_{N=1}^{s} a_{i_N j_s} = \min_{1\le j\le n} \sum_{N=1}^{s} a_{i_N j} = \underline{v}(s)\cdot s = \underline{v}s$, где $(i_1, j_1),...,(i_s, j_s)$ – пары стратегии игроков на *1, ..., s* ходах. А выбор $(s+1)$-го хода 1-го игрока, (после s ходов 2-го игрока) происходит выбором стратегии i_{s+1} согласно условию: $\sum_{N=1}^{s} a_{i_{s+1} j_N} = \max_{1\le i\le m} \sum_{N=1}^{s} a_{i j_N} = \overline{v}(s)\cdot s = \overline{v}s$. Цена игры v, как известно [10], удовлетворяет неравенству $\underline{v}(s) \le v \le \overline{v}(s)$.

Наглядно итерации метода изображены в табл. 1, где N – номер итерации, $A_1, A_2,..., A_m$ – соответственно стратегии первого, а $B_1, B_2,...B_n$ – второго игроков, $N\underline{v}$, $N\overline{v}$ – минимальное в столбцах B_j и максимальное в столбцах A_i на шаге N; в строку $\sum$ на каждой итерации записывается накопление за все итерации (платежи); $v^* = \dfrac{\underline{v}+\overline{v}}{2}$ – приближенное значение цены игры после $\boldsymbol{N}$ итераций.

Окончание работы алгоритма может производиться при 1 выполнении равенства $\underline{v}_{\max} = \overline{v}_{\min} = v$; 2) после выполнения определенного количества итераций, в таком случае находят $\max_{1\le N\le s} \underline{v}(N) = \underline{v}_{\max}$ и $\min_{1\le N\le s} \overline{v}(N) = \overline{v}_{\min}$. Если $\underline{v}_{\max} < \overline{v}_{\min}$, то можно брать $v = (\underline{v}_{\max} + \overline{v}_{\min})/2$.

Пример трех итераций решения игры с матрицей

$$A = \begin{pmatrix} 2 & 1 \\ 3 & 2 \\ 1 & 4 \end{pmatrix}$$

методом Брауна-Робинсон представлено в табл. 1.

Таблица 1 – Три итерации решения матричной игры методом Брауна-Робинсон

N	i	B_1	B_2	j	A_1	A_2	A_3	$N\underline{v}$	$\underline{v}$	v^*	$N\overline{v}$	$\overline{v}$
1	1	2	1	2	1	2	4	1	1	2.5	4	4
	$\sum$	2	1	$\sum$	1	2	4					
2	3	1	4	1	2	3	1	3	1.5	2	5	2.5
	$\sum$	3	5	$\sum$	3	5	5					
3	2	3	2	1	2	3	1	6	2	2.3	8	2.7
	$\sum$	6	7	$\sum$	5	8	6					

Рассмотрим другой итеративный процесс – монотонный итерационный алгоритм.

Монотонный итерационный алгоритм решения матричных игр

Опишем согласно [11] монотонный алгоритм как итерационный процесс, который позволяет найти оптимальное значение v цены игры для игры Γ_A, которая задана матрицей $A=\left(a_{ij}\right)$ размерности $m \times n$.

На нулевом шаге первый игрок выбирает произвольную чистую стратегию $x^0=\left(x_1,x_2,...,x_m\right)$ и определяется вспомогательный вектор c^0 как вектор скалярных произведений столбцов матрицы A и вектора x^0, то есть $c_j^0=\sum_{i=1}^m a_{ij}x_i$, $\forall j=\overline{1,n}$, $c=\left(c_1^0,...,c_n^0\right)$.

Шаг 1. Устанавливаем номер итерации N, равный 1, $N=1$.

Шаг 2. Определяем $\underline{v}^{N-1}=\min_{j\in J_n} c_j^{N-1}$ и обозначим $J^N=\left\{j_1^{N-1},...,j_\gamma^{N-1}\right\}$ множество индексов, на которых достигается $\underline{v}^{N-1}$, то есть $J^N=A\operatorname{rg}\min_{1\le j\le n} c_j^{N-1}$.

Шаг 3. Формируем матричную игру $\Gamma^N \subset \Gamma_A$ как частичную игру Γ_A с матрицей $A^N=\left(a_{ij}^N\right)$, где $i=\overline{1,m}$, $j\in J^N$.

Шаг 4. Решаем сформированную частичную игру и определяем оптимальную стратегию $\tilde{x}^N=\left(\tilde{x}_1^N,...,\tilde{x}_m^N\right)$.

Шаг 5. Определяем вектор $\tilde{c}^N=\left(\tilde{c}_1^N,...,\tilde{c}_n^N\right)$ следующим образом: $\tilde{c}_j^N=\sum_{i=1}^m a_{ij}\tilde{x}_i$, $\forall j=\overline{1,n}$.

Шаг 6. Рассматриваем игру с матрицей размерности $2\times n$: $\begin{bmatrix} c_1^{N-1},...,c_n^{N-1} \\ \tilde{c}_1^N,...,\tilde{c}_n^N \end{bmatrix}$.

Находим оптимальную стратегию $\left(\alpha_N,1-\alpha_N\right)$, $0\le\alpha_N\le 1$ первого игрока в этой игре. При этом α_N выбираем согласно следующему соотношению:

$$\alpha_N=\arg\max_{\alpha}\min_{j}\left(\left(1-\alpha\right)c_j^{N-1}+\alpha\tilde{c}_j^N\right).$$

Шаг 7. Если $\alpha_N=0$, то выход из алгоритма с ценой игры $\underline{v}^{N-1}$, иначе – переход на следующий шаг алгоритма.

Шаг 8. Вычисляем значение вектора $x^N=\left(x_1^N,...,x_m^N\right)$: $x^N=\left(1-\alpha_N\right)x^{N-1}+\alpha_N\tilde{x}^N$.

Шаг 9. Определяем компоненты вектора $c^N=\left(c_1^N,...,c_n^N\right)$ согласно формуле: $c^N=\left(1-\alpha_N\right)c^{N-1}+\alpha_N\tilde{c}^N$.

Шаг 10. Увеличиваем номер N итерации на 1, переходим на шаг 2 алгоритма.

Числовые эксперименты

С целью исследования практической эффективности рассмотренных методов для решения матричных игр и перспектив их применения в комбинаторной оптимизации для задач [1-9] они были программно реализованы на языке Delphi. С использованием этой программной реализации методов проведена серия числовых экспериментов на компьютере с процессором *AMD FX- 6300* с тактовой частотой 3.5 Ггц и объемом оперативной памяти 8 ГБ.

Результаты экспериментов приведены в табл. 2 - 4, в которых указана размерность задачи – параметры n и m – и время, которое было потрачено на вычисления (решалось по 10 задач при каждой размерности, приведено общее время вычислений всех задач), где «мс» – миллисекунды, «с» – секунды, «мин» – минуты, «ч» - часы. Элементы матрицы A для каждой задачи были сгенерированы как целые числа с равномерным распределением в интервале $[1;1000]$. Работа алгоритмов проводилась так:

1. Для метода Брауна-Робинсон до достижения относительной точности по цене игры вычислений $\varepsilon = 10E - 5$, т. е. при $\left|\underline{v}_{\max} + \overline{v}_{\min}\right| < \varepsilon$ осуществлялась остановка.

2. Для монотонного алгоритма до получения значения параметра $\alpha = 0$, что позволило получать точное решение задач. Следует отметить, что монотонный алгоритм требует решения на определенных этапах игровых задач размерности $n \times 2$, которые в данной программной реализации были решены с использованием симплекс-метода, что в некоторых случаях не позволило достичь максимальной эффективности алгоритма.

Для проверки точности решения были использованы решения этих же задач симплекс-методом, путем приведения игровых задач к паре двойственных задач линейного программирования.

Таблица 2 – Результаты числовых экспериментов (задачи типа $n \times 2$)

№ п/п	Размерность задач		Время вычислений		
	n	m	Метод Брауна-Робинсон	Монотонный итерационный алгоритм	Симплекс-метод
1.	1000	2	202 мс	68мс	16мс
2.	5000	2	702 мс	302мс	22мс
3.	10000	2	1с 631мс	618мс	38мс
4.	20000	2	2с 706мс	1с 231мс	63мс
5.	30000	2	5с 91мс	1с 897мс	86мс
6.	40000	2	6с 657мс	2с 593мс	145мс
7.	50000	2	6с 746мс	3с 237мс	178мс
8.	100000	2	16с 569мс	6с 415мс	415мс
9.	250000	2	42с 301мс	16с 672мс	1с 140мс
10.	500000	2	1мин 3с	33с 851мс	2с 498мс
11.	1000000	2	3мин 57с	1мин 30с	4с 924мс

Увеличение размерности задач в числовых экспериментах, до больших значений, чем те, что приведены в табл. 2 и 3, оказалось невозможным в связи с ограниченными вычислительными ресурсами компьютера (объемом оперативной памяти). В числовых экспериментах, результаты которых приведены в табл. 2, было решено ограничить размерность значениями $n = m = 1000$ в связи со значительным временем вычислений уже при этих параметрах.

С помощью математического пакета CurveExpert [13] была получена регрессионная зависимость времени работы алгоритмов от размерности задач и коэффициент корреляции R^2 (рис. 1-3).

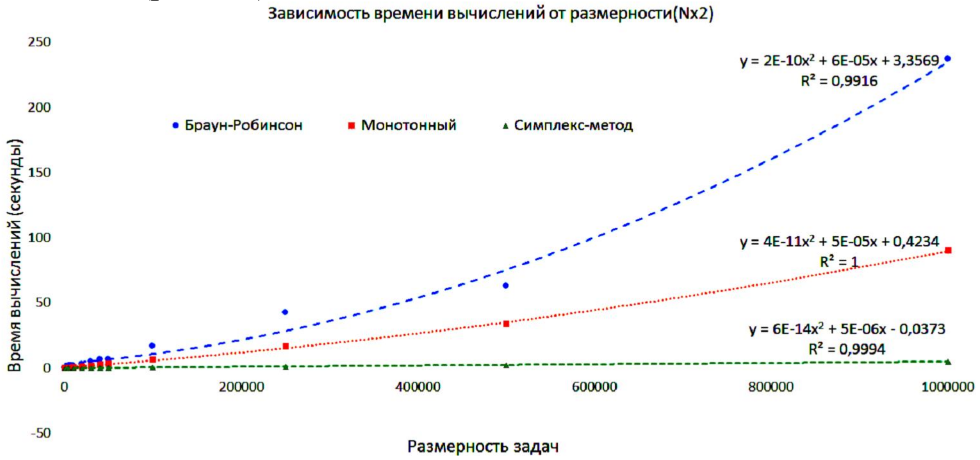


Рисунок 1 – Зависимость времени вычислений от размерности задачи (для случая $n \times 2$)

Таблица 3 – Результаты числовых экспериментов (задачи типа $n \times m$)

№ п/п	Размерность задач		Время вычислений		
	n	m	Метод Брауна-Робинсон	Монотонный итерационный алгоритм	Симплекс-метод
1.	10	10	1с 436мс	18мс	12мс
2.	20	20	7с 786мс	26мс	13мс
3.	30	30	21с 446мс	95мс	24мс
4.	40	40	38с 455мс	264мс	40мс
5.	50	50	1мин 29с	4с 317мс	84 мс
6.	100	100	4мин 59с	2мин 47с	1с 123мс
7.	200	200	20мин 5 с	16мин 20с	16с 182мс
8.	300	300	42мин 24с	38мин 17с	1мин 25с
9.	400	400	58мин 10с	1ч 5мин	4мин 10с
10.	500	500	1ч 40мин	1ч 58мин	9мин 50с
11.	600	600	2ч 10мин	3ч 40мин	24мин 10с
12.	700	700	2ч 57мин	5ч 30мин	42мин 20с
13.	800	800	3ч 50мин	7ч 30мин	1ч 56мин
14.	900	900	5ч 24мин	9ч 56мин	4ч 32мин
15	1000	1000	6ч 59мин	18ч 20мин	7ч 8мин

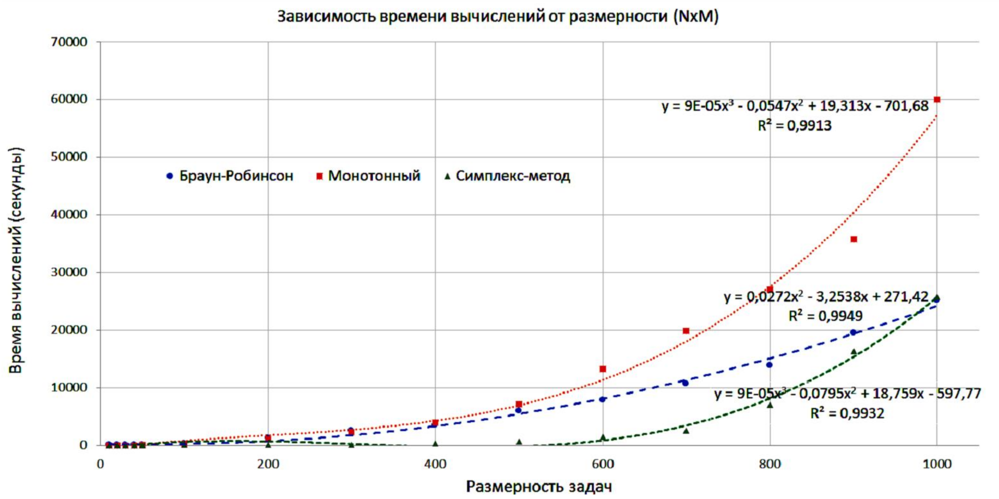


Рисунок 2 – Зависимость времени вычислений от размерности задачи (для случая $n \times m$)

Таблица 4 – Результаты числовых экспериментов (задачи типа $2 \times m$)

№ п/п	Размерность задач		Время вычислений		
	n	*m*	Метод Брауна-Робинсон	Монотонный итерационный алгоритм	Симплекс-метод
1.	2	100	892мс	35мс	19мс
2.	2	500	9с 634мс	582мс	244мс
3.	2	750	13с 836мс	1с 408мс	594мс
4.	2	1000	26с 492мс	11с 878мс	1с 309мс
5.	2	2000	1мин 4с	31с 109мс	5с 700мс
6.	2	3000	2мин 7с	57с 106мс	11с 721мс
7.	2	5000	4мин 33с	2мин 26с	36с 959мс
8.	2	10000	7мин 48с	5мин 44с	3мин 2с
9.	2	15000	14мин 28с	14мин 8с	6мин 34с
10.	2	20000	36мин 25с	29мин 11с	16мин 37с

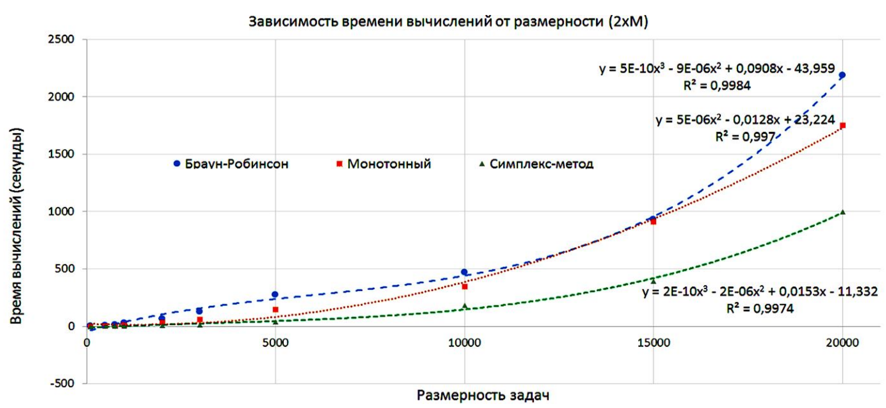


Рисунок 3 – Зависимость времени вычислений от размерности задачи (для случая $2 \times m$)

В результате проведенных числовых экспериментов выяснилось, что в случае размерности задач $2\times n$ и $n\times 2$ решение монотонным алгоритмом проводится за меньшее время, чем методом Брауна-Робинсон. При этом следует заметить, что монотонный алгоритм, в отличие от метода Брауна-Робинсон, позволил получить для каждой задачи точное решение, но при этом определить вероятности стратегий только одного игрока (в методе Брауна-Робинсон – обоих). Для задач с разными значениями m и n при размерности, большей 400×400, вычисления монотонным алгоритмом проводятся дольше, чем методом Брауна-Робинсон. Это может объясняться необходимостью проводить часть вычислений с использованием симплекс-метода, что обусловлено данной программной реализацией.

В ходе экспериментов выяснилось, что метод Брауна-Робинсон требует наименьший объем оперативной памяти, но при этом проводит значительно большее количество итераций. Также следует отметить, что в осуществленных числовых экспериментах решение задач симплекс-методом сведением матричной игры к паре двойственных задач линейного программирования, потребовало меньше всего времени, но в то же время использован значительный (наибольший из всех методов) объем оперативной памяти вычислительной системы.

Выводы

В статье приведены результаты числовых экспериментов по практической эффективности метода Брауна-Робинсон и монотонного алгоритма для матричных игр. В ходе проведенных исследований были проведены сравнения этих двух методов решения матричных игр с решением этих задач симплекс-методом с целью выяснения возможности их применения для решения задач комбинаторной оптимизации игрового типа. Полученные результаты позволяют сделать вывод, что монотонный алгоритм для задач большой размерности предпочтительнее, чем метод Брауна-Робинсон.

Как направление дальнейших исследований можно рассматривать разработку методов решения задач комбинаторной оптимизации игрового типа с использованием идей, на которых основывается монотонный алгоритм.

Список литературы

1. Емец О. А. Исследование математических моделей и методов решения задач на перестановках игрового типа / О. А. Емец, Н. Ю. Устьян // Кибернетика и сист. анализ. – 2007. – № 6. – С. 103-114.
2. Садовский А. Л. Монотонный итеративный алгормитм решения матричных игр. / А. Л. Садовский – ДАН СССР, 1978. – Т. 238, № 3. - С. 538-540.
3. Петросян Л.А. Теория игор./ Л.А. Петросян, Н.А. Зенкевич, Е.А. Семина. – М : Высш. шк., Книжный дом «Университет». – 1998. – 304 с.
4. Ємець О.О. Розв'язування ігрових задач на переставленнях / О.А. Емец, Н.Ю. Устьян // Наукові вісті НТУУ "КПІ". – 2007. – № 3. – С. 47-52.
5. Емец О.А. Решение некоторых задач комбинаторной оптимизации на размещениях и перестановках игрового типа / О.А. Емец, Н.Ю. Устьян. // Проблемы управления и информатики. – 2006. – № 3. – С. 37-47.
6. Емец О.А. Исследование задач комбинаторной оптимизации игрового типа на размещениях / О.А. Емец, Н. Ю. Устьян // Проблемы управления и информатики. – 2007. – № 1. – С. 26-36.
7. Ємець О.О. Один ітераційний метод розв'язування ігрових задач на переставленнях / О.А. Емец, Н.Ю. Устьян // Наукові вісті НТУУ "КПІ". – 2008. – № 3. – С.5-10.
8. Емец О.А. Игры с комбинаторными ограничениями / О.А. Емец, Н.Ю. Устьян // Кибернетика и сист. анализ. – 2008. – № 4. – С. 134-141.
9. Емец О.А. Итерационный метод решения комбинаторных оптимизационных задач игрового типа на размещениях / О.А.Емец, Е.В. Ольховская // Проблемы управления и информатики. – 2011. – № 3. – С. 69-78.

10. Ємець О.О. Розв'язування комбінаторних задач ігрового типу з обмеженнями-переставленнями у обох гравців: ітераційний метод / О.О. Ємець, О.В. Ольховська // Системні дослідження та інфораційні технології . – № 4. – С. 80-93.
11. Емец О.А. Доказательство сходимости итерационного метода решения задачи комбинаторной оптимизации игрового типа на размещениях / О.А. Емец, Е.В. Ольховская // Кибернетика и сист. анализ. – 2013. – № 1. – С. 102-114.
12. Julia Robinson An Iterative Method of Solving a Game / The Annals of Mathematics, Second Series. – Vol. 54, № 2 (Sep., 1951). – P. 296-301
13. CurveExpert Software / Daniel G. Hyams. – 2013. – Режим доступу : http://www.curveexpert.net/.

References

1. Yemets O. O. Study of mathematical models and methods of solving problems on permutations of the gaming type. Yemets O. O., Ustian N. Y. Kibernetika i sist. analiz (Ukraine), 2007, 6, pp. 103-114.
2. Sadovskiy A.L. Monotone iterative algorithm for solving matrix games. A.L. Sadovskiy. RAS USSR, 1978, vol. 238, № 3, pp. 538-540.
3. Petrosyan L.A. Games theory. L.A. Petrosyan, N.A. Tzenkevich, E.A. Semina. Moscow, 1998, 304 p.
4. Yemets O. O. Solving game problems on permutations. Yemets O. O., Ustian N. Y. Naukovi Visti NTUU "KPI" (Ukraine), 2007, 3, pp. 47-52.
5. Yemets O. O. Solving some combinatorial optimization problems on arrangements and permutations of the gaming type. Yemets O. O., Ustian N. Y. Problemy upravleniia i informatiki (Ukraine), 2006, 3, pp. 37-47.
6. Yemets O. O. Study of combinatorial optimization problems of the gaming type on arrangements. Yemets O. O., Ustian N. Y. Problemy upravleniia i informatiki (Ukraine), 2007, 1, pp. 26-36.
7. Yemets O. O. One iterative method of solving game problems on permutations. Yemets O. O., Ustian N. Y. Naukovi Visti NTUU "KPI" (Ukraine), 2008, 3, pp. 5-10.
8. Yemets O. O. Games with combinatorial restrictions. Yemets O. O., Ustian N. Y. Kibernetika i sist. analiz (Ukraine), 2008, 4, pp. 134-141.
9. Yemets O. A. The iterative method of solving combinatorial optimization problems of the gaming type on arrangements. Yemets O. O., Olkhovska O. V. Problemy upravleniya i informatiki (Ukraine), 2011, 3, pp. 69-78.
10. Yemets O. O. Solving combinatorial problems of the gaming type with permutations-restrictions of both players: the iterative method. Yemets O. O., Olkhovska O. V. Systemni doslidzhennia ta informatsiini tekhnolohii (Ukraine), 4, pp. 80-93.
11. Yemets O. O. Proof of convergence of the iterative method for solving combinatorial optimization problems of the gaming type on arrangements. Yemets O. O., Olkhovska O. V. Kibernetika i sist. analiz (Ukraine), 2013, 1, pp. 102-114.
12. Julia Robinson An Iterative Method of Solving a Game / The Annals of Mathematics, Second Series, Vol. 54, No. 2 (Sep., 1951), pp. 296-301
13. CurveExpert Software / Daniel G. Hyams. – 2013.

RESUME

O.O. Iemets, D.M. Olhovskiy, O.V. Olhovskaya

Comparison of Methods for Solving Game Tasks: Numerical Experiments

Background: in combinatorial optimization a new class of combinatorial optimization problems of the gaming type is considered, with two players, in which payments are presented as a matrix, and combinatorial restrictions are imposed on the strategies of the players. Such problems can be widely used in various fields of human activity. To solve this class of problems an approximate solving method is suggested, which is based on the same ideas as those being in the basis of the method of Brown- Robinson. It is well known that the method of Brown- Robinson has shown a certain practical efficiency, however in

separate classes of problems convergence speed was not sufficient. Taking this into consideration for solving combinatorial optimization game type it becomes necessary to develop other methods for solving matrix games, which will be based on other approaches.

Materials and methods: In our article we have set the task to compare the known methods for solving matrix games basing on numerical experiments, both in terms of speed of convergence and time-consumption, as well as in terms of their accuracy, in order to use this modification at solving combinatorial optimization problems of game type

Results: In order to investigate the practical effectiveness of the method of Brown-Robinson and monotonic algorithm for solving matrix games as well as the prospects for their use in combinatorial optimization , they were implemented in the language of Delphi. Using this software implementation of methods, a series of numerical experiments have been accomplished on a computer with an AMD FX 6300 processor with a clock speed of 3.5 GHz and 8 GB of RAM. In the course of the experiments 10 tasks have been solved in each dimension, the A matrix elements for each task were generated as integers with uniform distribution in the interval $[1;1000]$. To check the accuracy of the solution the simplex method has been used to solve the same problems, by bringing the game problems to a pair of dual problems of linear programming. With the help of mathematical package CurveExpert we have obtained regression dependence of the work time of algorithms on the dimension of the tasks and the correlation coefficient R^2, which showed that the time depends as a second-degree polynomial.

Conclusion: According to the results of experiments monotonic algorithm for large-dimension problems is preferred to the method of Brown- Robinson.

Статья поступила в редакцию 26.12.2013.